

# Comparative Study of Different Methods in Vibration-Based Terrain Classification for Wheeled Robots with Shock Absorbers

Mingliang Mei <sup>1,†</sup>, Ji Chang <sup>2,†</sup>, Yuling Li <sup>3</sup>, Zerui Li <sup>2</sup>, Xiaochuan Li <sup>4</sup> and Wenjun Lv <sup>2,\*</sup> 

<sup>1</sup> Department of Mechanical Engineering, Fujian Polytechnic of Information Technology, Fuzhou 350003, China; meimingliang@fjpit.edu.cn

<sup>2</sup> Department of Automation, University of Science and Technology of China, Hefei 230027, China; cjchange@mail.ustc.edu.cn (J.C.); lzerui@mail.ustc.edu.cn (Z.L.)

<sup>3</sup> School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China; ly195@mail.hfut.edu.cn

<sup>4</sup> Faculty of Technology, De Montfort University, Leicester LE1 9BH, UK; xiaochuan.li@dmu.ac.uk

\* Correspondence: wlv@ustc.edu.cn

† These authors contributed equally to this work.

Received: 23 January 2019; Accepted: XXX; Published: XXX

**Abstract:** Autonomous robots that operate in the field can enhance their security and efficiency by accurate terrain classification, which can be realized by means of robot-terrain interaction-generated vibration signals. In this paper, we explore the vibration-based terrain classification (VTC), in particular for a wheeled robot with shock absorbers. Because the vibration sensors are usually mounted on the main body of the robot, the vibration signals are dampened significantly, which results in the vibration signals collected on different terrains being more difficult to discriminate. Hence, the existing VTC methods applied to a robot with shock absorbers may degrade. The contributions are two-fold: (1) Several experiments are conducted to exhibit the performance of the existing feature-engineering and feature-learning classification methods; and (2) According to the long short-term memory (LSTM) network, we propose a one-dimensional convolutional LSTM (1DCL)-based VTC method to learn both spatial and temporal characteristics of the dampened vibration signals. The experiment results demonstrate that: (1) The feature-engineering methods, which are efficient in VTC of the robot without shock absorbers, are not so accurate in our project; meanwhile, the feature-learning methods are better choices; and (2) The 1DCL-based VTC method outperforms the conventional methods with an accuracy of 80.18%, which exceeds the second method (LSTM) by 8.23%.

**Keywords:** autonomous robots; terrain classification; vibration; LSTM

## 1. Introduction

The current decade has witnessed an increasing application of autonomous vehicles and mobile robots in industrial transportation, ground reconnaissance, planetary exploration, etc. [1,2]. These autonomous robots performing outdoor tasks must face various terrain types. Hence, an accurate terrain classification is of significant importance to prevent the robots from non-geometric hazards which are mainly rendered by slippery and bumpy surfaces. For example, when a field robot is trying to traverse a slippery terrain, the terrain-dependent slippage coefficients can be estimated by means of terrain classification; consequently, excessive wheel sinkage can be avoided by slippage compensation algorithms, i.e., torque-based or speed-based traction control strategies [3]. In addition, terrain classification has been demonstrated to have great contributions in many other aspects, i.e., adaptive

navigation, traction control, energy saving, gait control, etc. [4–8]. Therefore, the research on terrain classification have received great attention from the DARPA grand challenge and planetary exploration.

Robotic terrain classification refers to the process of a mobile robot identifying the current or forthcoming terrain, such as gravel, grass, etc., using its on-board sensors. The terrain classification methods can be categorized into two parts: the non-interactive and the interactive methods. The non-interactive methods are usually realized by optical sensors (e.g., digital camera, infrared camera and LiDAR) [9–12], which have been widely investigated. The non-interactive methods usually possess high classification accuracy, but they are demanding to environmental conditions (such as illumination), variations in appearance or cover (such as leaves). Alternatively, the interactive methods, which can be realized by means of acoustics [13–15], haptics [16–18] and vibration [19–21], are robust to changes in appearance and illumination, and have better computational efficiency. Therefore, the interactive methods have shown increasing potential in robotic environment perception. The acoustic terrain classification has been seldom studied due to a fatal issue of external environmental and internal mechanical noise [22]. The haptic terrain classification leverages ground reaction forces to recognize terrain, by means of tactile sensor arrays mounted on the robot-terrain contact area; therefore, they are more applicable to legged robots than wheeled robots [23]. Apart from the acoustic and haptic signal, vibration signal also provides sufficient information to discriminate different terrains based on mechanical properties [24]. In contrast to acoustic signal, the interference in accelerometer-gathered vibration signal is caused by gravitational acceleration, which is time-invariant and easy to eliminate. Additionally, vibration-based methods can be applied to both wheeled and legged robots compared with haptic ones. Considering the above advantages of vibration signal, we focus on VTC in this paper.

All the existing work concerning VTC concentrates on the robots with relatively hard tires and without shock absorbers, because the vibration signal could reflect the truth of ground surfaces. However, in real-world applications, to guarantee the stability and security of the robot itself, and protect the equipment or cargo on the robot, the robot is often equipped with elastic tires and shock absorbers. Hence, it is quite practical to study the VTC under the effect of shock absorbers. The difference between the vibration signals with and without shock absorbers can be seen in Figure 1. Obviously, the shock absorbers result in a significant dampening of vibration magnitude, especially the high frequency component, so the dampened vibration signal cannot reflect the fine surface characteristics. This means the terrains with small differences are difficult to discriminate from the dampened vibration signal. To the best of our knowledge, VTC problem under shock absorbers has not been investigated. Therefore, whether the existing relevant methods are applicable to dampened vibration signals should be verified. If not, more effective methods should be developed.

In this paper, we explore the VTC, in particular for wheeled robots with shock absorbers. Because the vibration sensors are usually mounted on the main body of the robot, the vibration magnitude are dampened significantly, which results in the vibration signals collected on different terrains being more difficult to discriminate. Hence, the existing VTC methods applied to a robot with shock absorbers degrades. The contributions are twofold:

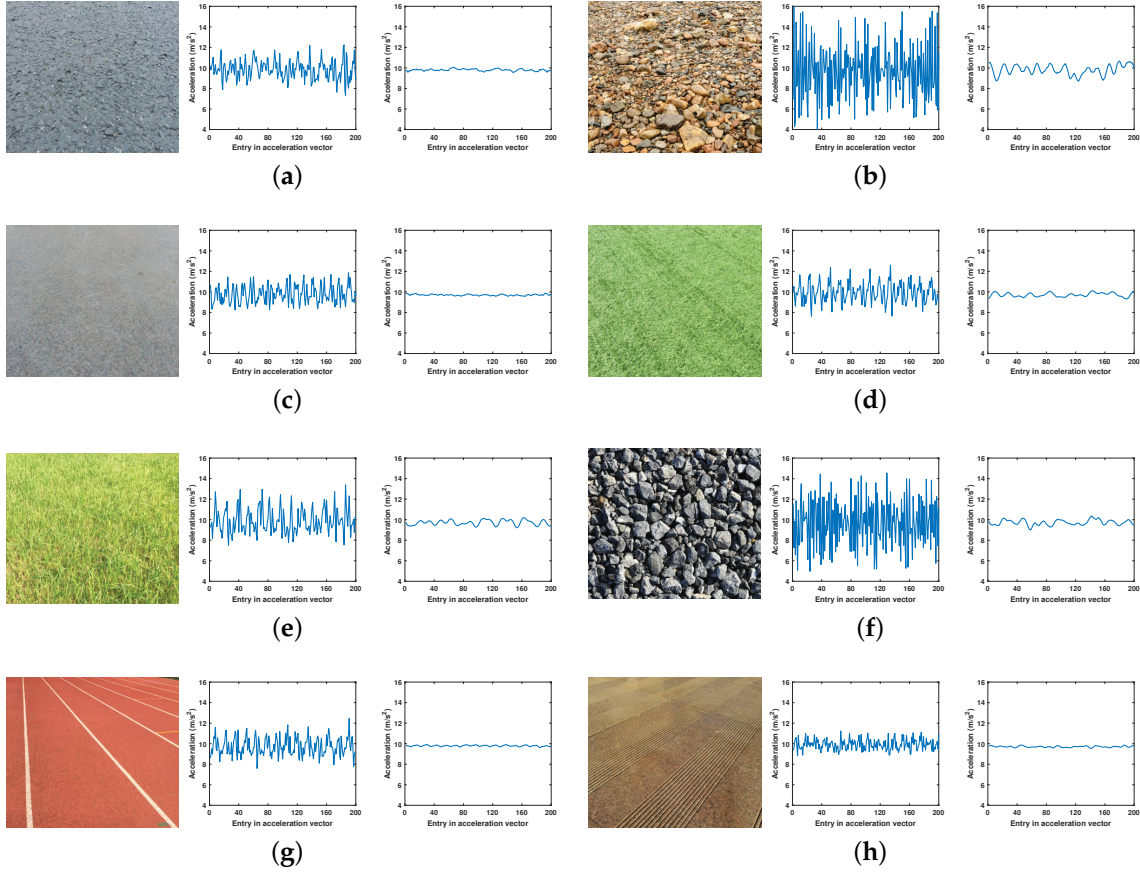
1. A large body of comparative experiments are conducted to demonstrate the performance of the existing feature-engineering and feature-learning classification methods.
2. According to the long short-term memory (LSTM) network, we propose a one-dimensional convolutional LSTM (1DCL)-based VTC method to learn both spatial and temporal characteristics of the dampened vibration signals.

The real-world experiment on eight different terrains demonstrates that:

1. In feature-engineering approaches, the combination of power spectral density (PSD) and random forest (RF) achieves the highest accuracy of 69.88%.
2. The conventional feature-learning approaches outperforms the feature-engineering ones slightly by about 2%, though they are computationally intensive.

3. The proposed 1DCL-based VTC method outperforms the conventional methods in an accuracy of 80.18%, which exceeds the second method (LSTM) by 8.23%.

The remainder of this paper is organized as follows. Section 2 presents related work. In Section 3, we describe the framework of this paper and summarize the feature-engineering approaches. In Section 4, we give a brief introduction of feature-learning approaches and describe our approach in detail. The dataset, evaluation metrics and experiment results are then presented in Section 5. Finally, in Section 6, we draw conclusions and give proposals for future work.



**Figure 1.** Examples of eight terrain types along with corresponding samples of undampened vibration signals (**left**) and dampened vibration signals (**right**). (**a**) Asphalt. (**b**) Cobble. (**c**) Concrete. (**d**) Grass 1 (artificial grassland). (**e**) Grass 2 (natural grassland). (**f**) Gravel. (**g**) Plastic. (**h**) Tile.

## 2. Related Work

A large body of studies about non-interactive methods using exteroceptive modalities have been done. Image-based methods usually use digital cameras or infrared cameras. Digital image-based methods mainly extract visual features from terrain images using local binary patterns (LBP), bag of visual words (BOVW) or speeded up robust features (SURF) [10,11,25,26]. Recently, convolutional neural networks (CNNs) have attracted more attention due to their excellent feature extracting capability, and thus applying to near-range and far-range terrain classification and planetary surface soil property analysis [27,28]. Digital cameras may not perform well in the dark. In contrast, infrared cameras which usually cooperate with laser or ladar can work at night [29]. However, infrared image-based or ladar-based methods focus on segmenting the ground surface from different obstacles [30–32], rather than classifying the terrain type.

Interactive methods using proprioceptive modalities has not been explored in the same depth as non-interactive methods. The existing interactive methods are usually realized by acoustics, haptics,

and vibration. Acoustic terrain classification is motivated by vehicle-terrain interaction sounds which is produced when the robotic locomotion mechanism (e.g., legs, wheels or tracks) is pressing the ground. This idea is first proposed by [22]. In their work, the acoustic features are extracted by surveying acoustic methods from other domains and their efficacy is verified in terrain classification. Adopting feature extraction methods from speech processing literature, a novel feature vector composed of zero crossing rate, spectral band energies and their vector time derivatives is proposed [13]. In [15], a deep spatiotemporal model is designed for learning complex dynamics in audio signals which replaces manually handcrafted features. Their model yields an average classification accuracy of 97.36% on the offline experiment. The acoustic terrain classification has a fatal issue of external environmental and internal mechanical noise, so it is impractical in noisy outdoor environment. One alternative modality, haptics, is most investigated on legged robots. Haptic feedback is obtained by force sensors or capacitive tactile sensors mounted on robotic legs [16,23]. According to the sensory information and the perceived terrain type, legged robots can perform gaits and speed adjustment for more effective motion. More work can be found in the literature [17,18].

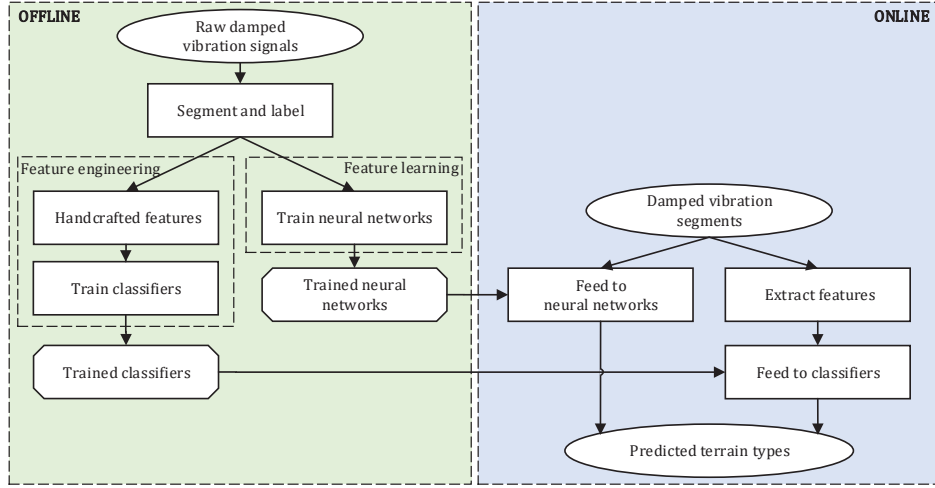
The most popular proprioceptive modality in interactive method is vibration. Vibration signals are easily collected in contrast to haptic signals, since the classifier can perform well even with a consumer-grade accelerometer. The early work that involves the VTC uses power spectrum as the feature [33]. In later work, researchers begin to develop more simple and compact features in the time domain instead of a more than 100-dimension frequency-domain feature [34]. An 8-dimensional feature vector composed of zero crossing rate, mean, standard deviation, etc., is then developed. This method costs less computing time than the spectral method; however, in some applications, these compact features are not sufficient to cover all the characteristics of the raw data. Support Vector Machine (SVM) has been demonstrated to be the best performing classifier using time- and frequency-domain features [21,34]. In [20], Dynamic Cortex Memory (DCM) which is an extension of LSTM is addressed for VTC without any explicit feature computation. Their experiments on 14 terrain types achieve an overall accuracy of approximately 85%, which is the state-of-the-art accuracy for 14-class terrain classification. Moreover, they implement just-in-time computation for their networks. Apart from terrain classification, vibration is also frequently used in damage detection field [35–37]. In their work, some use time-domain features with a specific goal, such as kurtosis, crest factor and Root Mean Square (RMS), and others use Discrete Fourier Transform (DFT) to obtain frequency-domain features. These literatures involved feature extraction methods and classification algorithms will appear in the comparative study.

Multi-modality methods combine several modalities based on their complementary characteristics, thus possessing a higher accuracy and robustness against environmental interference [38,39]. In [40], three-axis acceleration, roll-pitch-yaw (RPY), angular rates, and cloud point images are combined by using bagging algorithm. The results achieve a similar level of accuracy as SVM and visual sensor data, with less computation. In [41], five data sources including four vibration sources and one acoustic source are collected by their tracked robot for fusing predictions. A two-stage feature selection method that combines Relief and mRMR algorithms is developed to obtain optimal feature subsets. In addition, four different classifiers are combined for classification task.

In the following comparative study, we refer to some of the feature extraction methods and classification methods of the aforementioned literature.

### 3. Feature-Engineering Approaches

The dampened VTC is illustrated in Figure 2. The entire process is divided into the offline training and the online classification. In the offline process, the raw dampened vibration signals are used to train the classifiers; in the online process, the trained classifiers are used to predict the testing dampened vibration segments.



**Figure 2.** Illustration of dampened vibration-based terrain classification.

In this section, we introduce the feature engineering approaches.

### 3.1. Vibration Feature Extraction

We collected vibration data by an accelerometer mounted perpendicularly on the main body of a wheeled mobile robot with shock absorbers. In our experiment, we recover the vibration signal from the accelerometer readings by subtracting the gravitational acceleration, and then split the vibration signal into short segments. Each segment contains 50% overlap of successive segments. In Section 5, we evaluate different segment sizes that affect the classification accuracy.

Now we are in the position to extract features from each segment. Feature extraction is performed in the time domain and in the frequency domain. Time-domain vibration feature extraction method aims to obtain a simple and compact representation of vibration signal. In VTC, mean, standard deviation, norm, autocorrelation, RMS, maximum and minimum are usually selected as statistical features [34,42]. In vibration-based damage detection, RMS, skewness and kurtosis are used since they have been proved to be useful for bearing fault detection [35,43]. Acoustic terrain classification is similar to the vibration-based one, so we refer to its features such as zero crossing rate and short time energy [13,14,22]. Here, we summarize the most common time-domain features.

Consider the training sample set  $D = \{v_1, v_2, \dots, v_m\}$ , where  $m$  is the number of samples. For each sample  $v_j = (v_{j,1}, v_{j,2}, \dots, v_{j,n}) \in D$ , where  $n$  is the number of acceleration values in a segment, the time-domain features are the followings. Please note that we will omit subscript  $j$  later for brief. That is to say, symbol  $v_i$  means the  $i$ th value of sample  $v$ .

- (1) Zero crossing rate (ZCR). The ZCR counts the number of times that the signal crosses the zero axis, which is an approximate estimation of the frequency of  $v$ .
- (2) Mean. The mean expresses the average roughness of ground surface.
- (3) Standard deviation. Intuitively, the standard deviation is greater with a rougher ground surface.
- (4) Norm. Usually the  $\ell_2$ -norm is used which reflects the energy of  $v$ .
- (5) Autocorrelation. The autocorrelation  $r$  is a measure of the non-randomness as defined by

$$r = \frac{1}{n\sigma^2} \left( \sum_{i=1}^{n-1} (v_i - \mu)(v_{i+1} - \mu) \right), \quad (1)$$

where  $r$  gets larger with the dependence of successive vibration values.

- (6) Maximum. Find the maximum of the degree of bump.
- (7) Minimum. Find the minimum of the degree of bump.



- (8) Skewness. The skewness  $Sk$  describes the asymmetry of the distribution about its mean, calculated as

$$Sk = \frac{[\frac{1}{n} \sum_{i=1}^n (v_i - \mu)^3]}{\sigma^3}. \quad (2)$$

- (9) Excess kurtosis. The excess kurtosis  $EK$  reflects the degree of deviation from the Gaussian distribution as defined by

$$EK = \frac{1}{n} \frac{[\frac{1}{n} \sum_{i=1}^n (v_i - \mu)^4]}{\sigma^4} - 3. \quad (3)$$

Frequency-domain representation is widely used in signal processing field as it simplifies the mathematical analysis and helps analyze the components of signals. The most common tool for frequency-domain transformation is Fast Fourier Transform (FFT) that is an extensive application in both VTC [44,45] and acoustics-based terrain classification [15,22]. PSD that describes how power of a signal is distributed over frequency is also used for processing vibration signals [33,42]. In this paper, we perform both FFT and PSD to convert vibration signals from time domain into frequency domain. FFT is an efficient computing algorithm for DFT. Given a sequence  $x(i)$  of length  $n$ , the  $N$ -point DFT of  $x(i)$  is defined as

$$X(k) = \mathcal{D}\{x(i)\}_N = \sum_{i=0}^{N-1} x(i)e^{-j\frac{2\pi}{N}ki}, k = 0, 1, \dots, N-1. \quad (4)$$

Usually  $N \geq n$  and  $x(i)$  is zero-padded which means the terms from  $a_{n+1}$  to  $a_N$  are padded with zero. Typically,  $N$  is specified as a power of 2. Hence, the training vectors of different lengths in our experiments implement the corresponding points FFT. PSD is defined as the Fourier transform of the autocorrelation function. For limited discrete values, the PSD of  $X(i)$  denoted by  $S_X(\omega)$  is estimated by

$$S_X(\omega) = \mathcal{D}\{R_X(k)\} = \sum_{k=-N}^N R_X(k)e^{-j\omega k} \quad (5)$$

where  $R_X(k)$  denotes the autocorrelation function of  $X(t)$ . A log scaling of the PSD is implemented in this study to reduce the dominating effect of high-magnitude frequency components.

### 3.2. Classifiers

Various classifiers have been used for VTC and other fields. Among them, SVM is used the most widely [21,34,39,45]. Other common classifiers such as k-nearest neighbor (kNN), decision tree (DT), Naïve Bayes (NB) and extreme learning machine (ELM) are also applied [40,44,46]. Additionally, ensemble learning, e.g., RF and AdaBoost, is a research hotspot in machine learning for its superior generalization performance [10,47]. In our experiment, we adopt seven classifiers: SVM [48], ELM [49], kNN (Classic kNN determines the class by the majority vote, which suffers from sample imbalance. Therefore, instead of using classic kNN, we employ a centroid displacement-based k-nearest neighbors algorithm [50] which has been proved to be adaptive to noise and class distributions) [50], NB [51], DT [52], RF [53] and AdaBoost [54]. KNN is a benchmark method in our study.

## 4. Feature-Learning Approaches

The classification performance of feature-engineering approaches relies on handcrafted feature extraction heavily. In recent years, there has been a considerable effort on the development of end-to-end learning methods [55]. Instead of manually extracting characteristic features, end-to-end learning method can learn the discriminative feature representation directly from raw data. The latter approach does not require too much prior knowledge of the problem or human expertise, and is advantageous in tasks where some high-level, abstract features from raw data are almost impossible to be developed manually. Usually, end-to-end learning method like deep neural network suffers from

computationally intensive training process. However, once the network is trained, it can be directly assembled to the mobile robots, thus not computationally intensive.

In this section, we first introduce two widely used end-to-end feature-learning methods: LSTM and CNN. Next, we propose a novel neural network architecture which is called 1DCL.

#### 4.1. Overview of CNN and LSTM

##### 4.1.1. Convolutional Neural Network

A typical convolutional neural network consists of alternating convolution and subsampling layers after the input layer and a generic multilayer network (fully connected layers) at the last stage of the architecture. At a convolution layer, the input feature maps from the previous layer are convolved with kernels which should be learned, and then the convolved results are put through the activation function to form the output feature map. After feature extraction in the convolution layers, the output feature map is transferred to subsampling layers (also named pooling layers) for feature selection and information filtering. The last few layers are fully connected layers that are identical to a multilayer perceptron (MLP) for estimating the decision (classification) vector. In CNNs, each convolutional neuron is only connected to some of the nodes in the previous layer called receptive field, which is a main characteristic of CNNs named sparse connection. Another characteristic of CNNs is parameter sharing that all the units in a feature map share the same parameters. These parameters constituted matrix refers to as kernel.

##### 4.1.2. Long Short-Term Memory

Recurrent neural networks (RNNs) can process and predict sequence data due to their recurrent structure [20]. However, traditional RNNs are unable to solve long-term dependencies problem and often suffer from vanishing and exploding gradient problems during training. The LSTM architecture [56] can overcome vanishing gradient problem effectively and enables exploitation of long-term temporal dynamics of a sequence.

The LSTM contains special units called memory blocks (mentioned as LSTM cells in the following) in the recurrent hidden layer. The LSTM cells contain a memory cell  $c_t$  used to store the temporal state of the network and three gates (at each time step  $t$ ) called forget gate  $f_t$ , input gate  $i_t$  and output gate  $o_t$ , respectively. The forget gate controls the forgotten states of the previous LSTM cell, the input gate controls the input activations into the LSTM cell, and the output gate controls the output of cell activations. The LSTM transition equations are the following:

$$\begin{aligned}
 i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \\
 f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \\
 o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \\
 u_t &= \tanh(W_{ux}x_t + W_{uh}h_{t-1} + b_u) \\
 c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned} \tag{6}$$

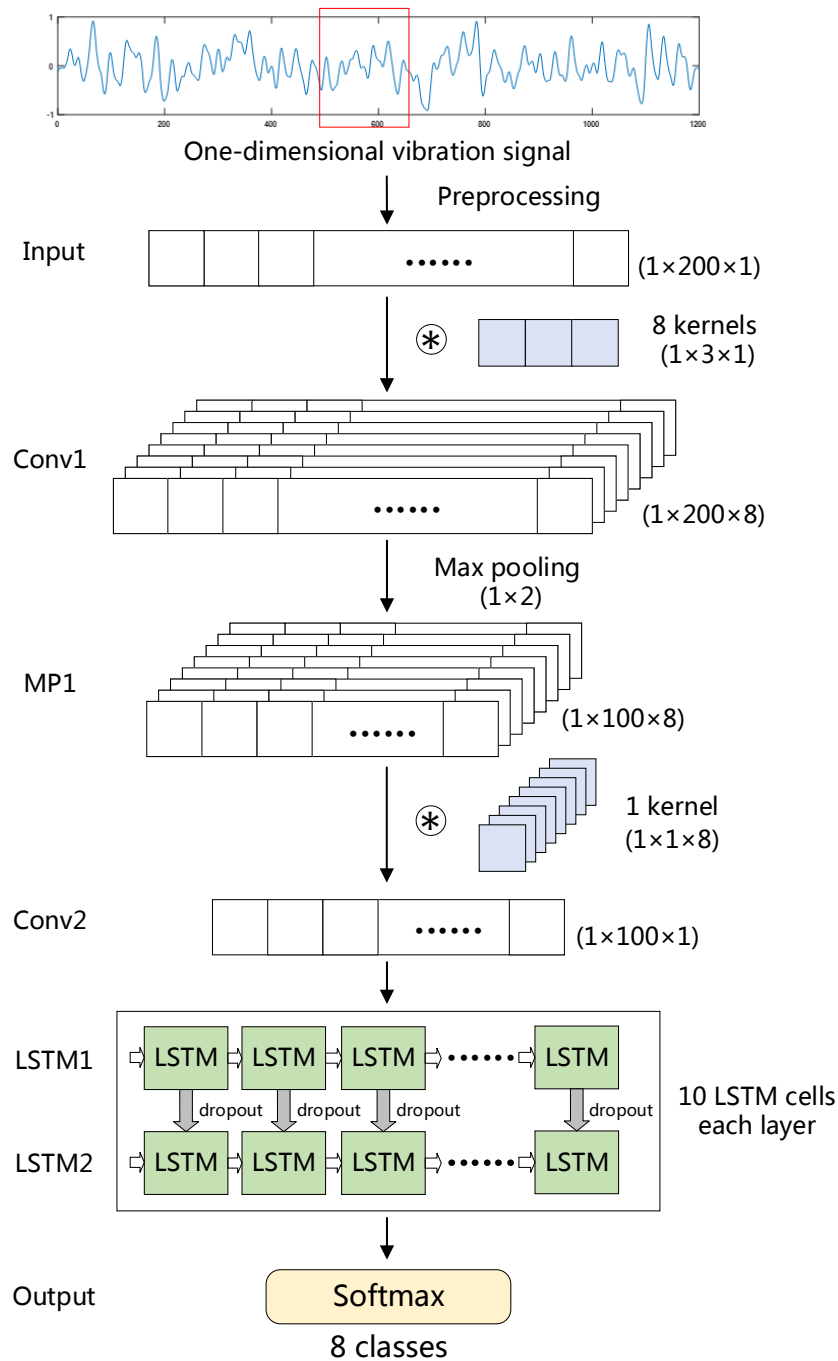
where  $x_t$  is the current input vector,  $h_t$  is the current cell output activation vector,  $u_t$  is the candidate value,  $W$  and  $b$  are weight vectors and bias vectors.  $\sigma$  and  $\tanh$  denote the logistic sigmoid function and the hyperbolic tangent function, respectively.  $\odot$  denotes element-wise multiplication.

#### 4.2. Proposed 1DCL

To our knowledge, the CNN-based and LSTM-based VTC have rarely been investigated. Many existing CNN and LSTM models are often general-purpose, thus not suitable for our application. In our study, the dampened vibration signals increase the difficulty of terrain classification, so we

design a dedicated neural network model called 1DCL by modifying and integrating CNN and LSTM. Such a neural network model can learn both spatial and temporal characteristics of the raw vibration signals.

The proposed 1DCL shown in Figure 3 consists of a 1D convolutional layer connecting with a max pooling layer, a  $1 \times 1$  convolutional layer and a two-layer LSTM and a SoftMax layer. Before fed into the 1DCL, the input vibration signals are preprocessed. Just as with previous operations, the raw vibration signals are split into short segments with 50% overlap of successive segments, and each segment contains 200 acceleration values. In addition, we normalized the segments into zero mean and unit standard deviation, which is a common procedure for neural networks that usually leads to the best classification performance.



**Figure 3.** The overall structure of 1DCL. The marker  $\otimes$  denotes convolution operation.



We employ a 1D convolutional layer in the 1DCL to learn preliminary spatial features. Structural differences between the traditional 2D and the proposed 1D CNN are visible, that is, the usage of 1-D arrays replace 2-D matrices for both kernels and feature maps. In the 1DCL, convolution kernel of the 1D convolutional layer is of size 3. Convolutional stride is fixed as 1. A max pooling layer with a kernel of size 2 follows the 1D convolutional layer. Max pooling layer provides some invariance and reduces parameters, thereby being able to avoid overfitting. The  $1 \times 1$  convolution which is first proposed in [57] can change the number of the channels of feature maps without changing the length and width of feature maps. Here we employ a  $1 \times 1$  convolutional layer after the convolutional and pooling layer, to reduce the number of channels of feature map into 1 (i.e., dimension of  $1 \times 100 \times 1$ ). In the two convolutional layers, we applied rectified linear unit (ReLU) as the nonlinear activation function. The ReLU has been proved to solve the vanishing gradient problem. Besides, it expedites the convergence of the training procedure compared with sigmoid and tanh activation function [58].

Rather than constructing multiple convolutional layers to learn deep spatial features, we only build the above two convolutional layers because we pay more attention to learn temporal dynamics from the vibration sequence. As shown in Figure 3, a two-layer LSTM is stacked after the  $1 \times 1$  convolutional layer. At each time step of the feature map, the outputs of the previous LSTM cell ( $c_t$  and  $h_t$  in Equation (6)) are used as the inputs of the LSTM cell at next time step. The output activation vector of the LSTM cell  $h_t$  in the first layer is used as the input of the LSTM cell in the second layer. For each layer, there are ten cells. Dropout regularization is used between the two LSTM layers to avoid overfitting with dropout probability of 0.5. Finally, a SoftMax layer is used to output the classification results.

As for the hyperparameters, we adopt the Spearmint Bayesian optimization library [59] to tune the hyperparameters in the artificial neural network, and grid search combined with 5-fold cross-validation to tune the hyperparameters in the feature-engineering approaches (e.g., SVM, ELM).

## 5. Experiment and Results Analysis

To compare different classification methods applied in terrain classification, we conduct the experiment with a four-wheeled mobile robot on 8 different terrains.

### 5.1. Experiment Setup

The experimental robot is shown in Figure 4. The robot is equipped with 4 dampers and 4 elastic tires, which constitute the robotic shock absorber. An IMU sensor is mounted on the robot roof to perceive the damped vibration.

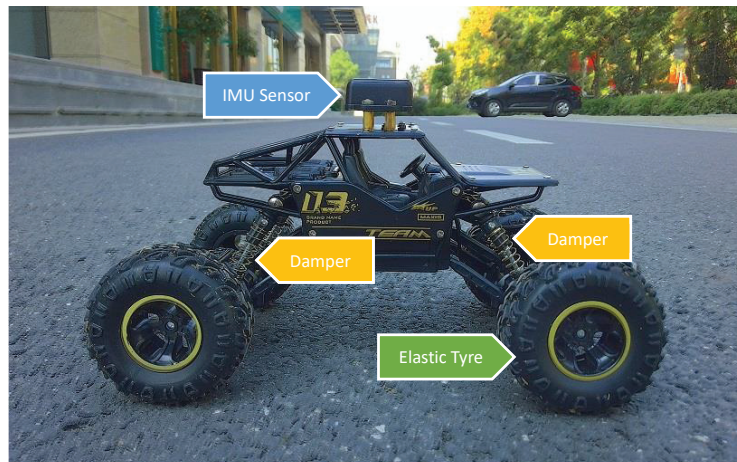


Figure 4. The experimental robot.

We collect vibration data by controlling the four-wheeled mobile robot to wander on eight terrain types which are different in rigidity, roughness, and flatness. Some of them are artificial terrains (e.g., asphalt, artificial grassland), while some are natural ones (e.g., cobble, natural grassland). In our experiment, we recover the vibration signal from the accelerometer readings by subtracting the gravitational acceleration, and then split the vibration signal into short segments. The length of each segment is set as 200 points initially, and each segment contains 50% overlap of successive segments; consequently, the dataset contains 11,224 samples in total (Asphalt: 1241; Cobble: 1434; Concrete: 1322; Artificial grassland: 1434; Natural grassland: 1562; Gravel: 1222; Plastic: 1624; Tile: 1385). Figure 1 shows the photos of the eight terrain types, along with the corresponding samples of undampened vibration signals and dampened vibration signals. Obviously, the shock absorbers result in a significant dampening of vibration magnitude, especially the high frequency component, so the dampened vibration signal cannot reflect the fine surface characteristics. The experimental robot traverses the eight terrains at a speed varying between 0.7 and 1.1 m/s, and in different motion mode (e.g., circular and linear motion) to prevent the classifiers from overfitting a certain motion.

## 5.2. Experiment Results and Analysis

In this section, we first study the performance of different classifiers in combination with handcrafted vibration features, i.e., feature-engineering approaches. Each pair of handcrafted feature and classifier is evaluated to find out the best combination. We also analyze the performance of 1DCL and compare it with feature-engineering approaches and other feature-learning approaches. Finally, experiments on different segment lengths are done.

The 10-fold cross-validation is used to evaluate the classification performance by three parameters, namely, accuracy, True Positive Rate (TPR) and F1-score. The accuracy  $Ac$ , which indicates the overall capability of the classifier to identify different terrains correctly, is given as

$$Ac = \frac{N_c}{N_t} \quad (7)$$

where  $N_c, N_t$  denote the number of correctly classified samples and the total number of the testing set, respectively. We focus on not only the overall capability of a classifier, but also the discriminability of each class which are evaluated by the indices of precision and recall. The precision  $Pr$  evaluates the correct rate predicted by the classifier for a certain class, and the recall  $Rc$  evaluates the proportion of correctly classified samples in class of interest, as follows

$$Pr = \frac{TP}{TP + FP'} \quad (8a)$$

$$Rc = \frac{TP}{TP + FN'} \quad (8b)$$

where the definitions of  $TP, FP$  and  $FN$  are illustrated in Figure 5. The TPR has the same meaning as the recall, so it is also calculated by Equation (8b). The F1-score takes both precision and recall into account.

We have

$$F1 = \frac{2 \times Pr \times Rc}{Pr + Rc}. \quad (9)$$

The calculation of  $Pr, Rc$ , and  $F1$  shown above are used in binary classification. We can simply apply the macro-averaging technology [60], i.e., averaging  $Pr, Rc$ , and  $F1$  of each class, to extend these metrics in multi-class classification.

		Predicted Results	
		Positive	Negative
True Condition	Positive	TP(True Positive)	FN(False Negative)
	Negative	FP(False Positive)	TN(True Negative)

**Figure 5.** Confusion matrix of binary classification. The confusion matrix of multi-class classification can be reduced to several confusion matrices of binary classification.

### 5.2.1. Analysis of Performance of Feature-Engineering Approaches

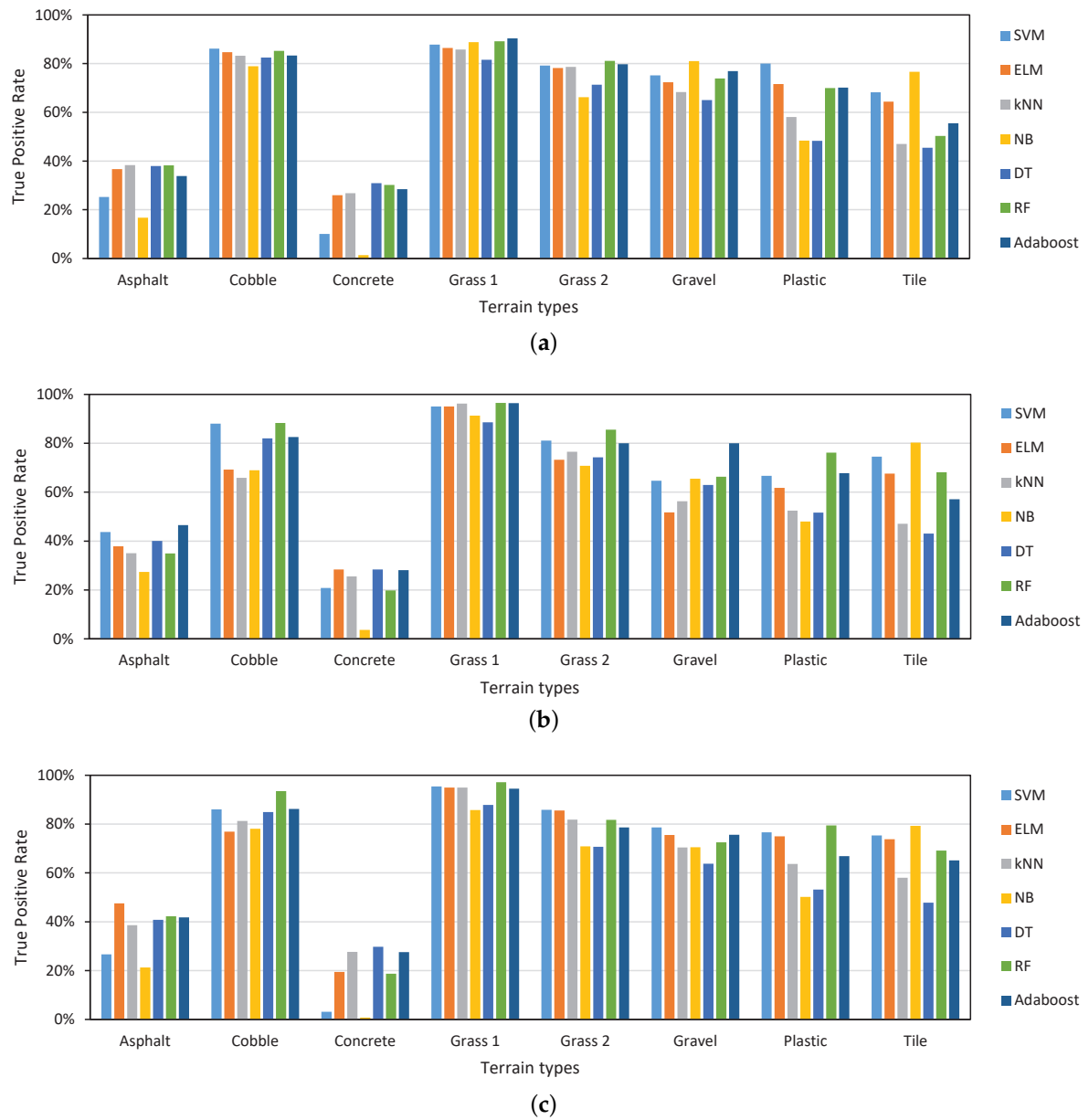
Table 1 shows the accuracy and F1-score of seven different classifiers using time-domain and freq-domain (including FFT-based and PSD-based) feature extraction methods. In general, random forest classifier using PSD-based features (PSD-RF for short) achieves the best performance of 69.88% accuracy and 0.6897 F1-score. With respect to the features, PSD is the best description of the characteristics of vibration signals overall, followed by FFT. Time-domain statistical features seem not to express the essential characteristics very well. One possible reason for the difference between freq-domain and time-domain features is that the complexity and irregularity of undampened vibration signals in time domain make it difficult to describe the most distinguishing characteristics. However, when transformed to frequency domain, complex vibration signals are decomposed into several single harmonic components, which facilitates the description of the characteristics. As for the classifiers, the performance of AdaBoost, ELM, RF, and SVM are similar, outperforming the other three classifiers significantly. SVM was found to be the most accurate method for vibration-based classification in [21,34], while ELM have been proved to have good generalization capability [49]. Naïve Bayes and DT have weak learning ability for our dampened vibration data. Their performance are even worse than the benchmark kNN in terms of time-domain features and PSD-based features. Results can be improved through some ensemble approaches, e.g., RF and AdaBoost. As shown in Table 1, RF and AdaBoost based on weak learner of DTs improve the prediction accuracy by over 7.5%.

**Table 1.** Performance of Seven Different Classifiers Using Time-Domain and Freq-Domain Feature Extraction Methods.

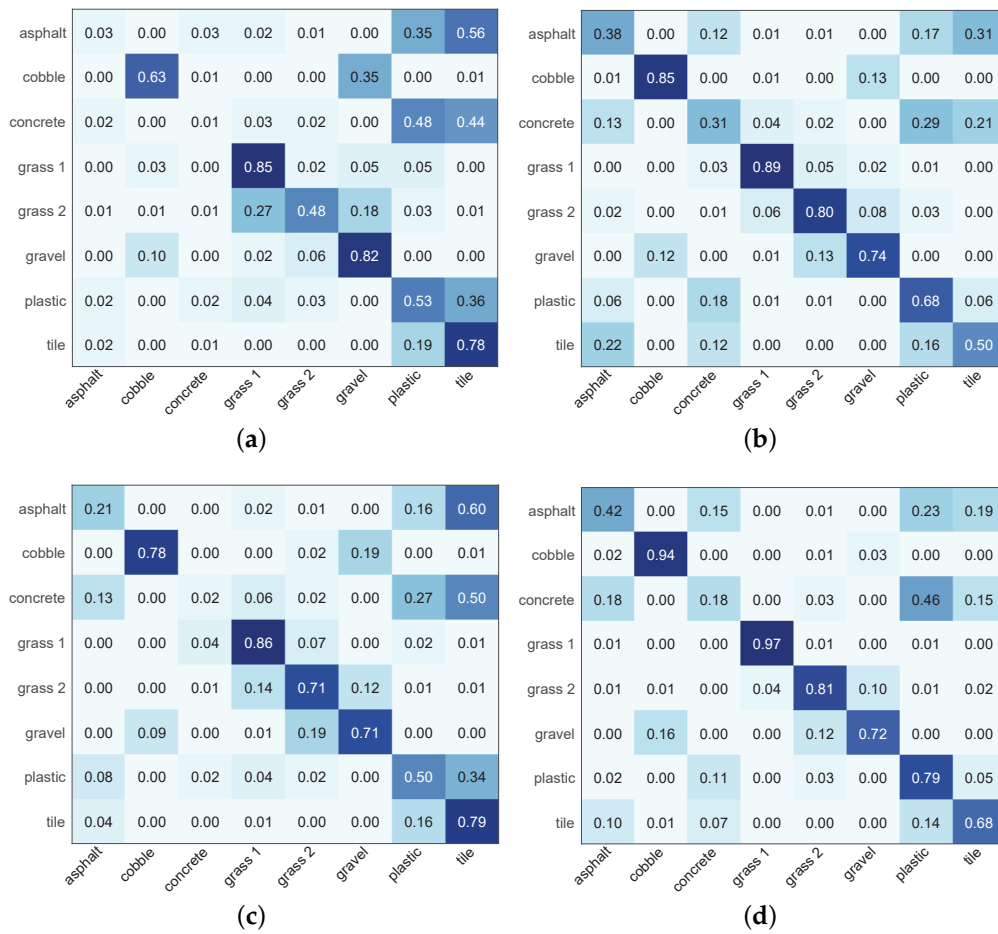
Features	Metrics	Classifiers						
		SVM	ELM	kNN	NB	DT	RF	Adaboost
Time-domain	Accuracy	65.30%	65.95%	61.53%	57.96%	58.39%	65.89%	<b>66.07%</b>
	F1-score	0.6438	<b>0.6497</b>	0.6065	0.5617	0.5803	0.6475	0.6465
FFT-based	Accuracy	67.99%	62.96%	57.75%	57.67%	59.62%	68.07%	<b>68.12%</b>
	F1-score	0.6674	0.6068	0.5695	0.5612	0.5898	0.6679	<b>0.6717</b>
PSD-based	Accuracy	67.52%	69.84%	65.69%	58.37%	60.26%	<b>69.88%</b>	67.91%
	F1-score	0.6622	0.6853	0.6454	0.5595	0.5999	<b>0.6897</b>	0.6699

Considering the ability to distinguish each terrain type, as illustrated in Figure 6, the TPR for grass 1 of each classifier is the highest overall, which demonstrates that grass 1 is the easiest to identify. For cobble, grass 2, gravel, plastic, and tile, the TPR is also relatively high. However, for asphalt and concrete, the TPR is always low. To further investigate the terrain types to which asphalt and concrete are misclassified, Figure 7 presents the confusion matrices of RF and NB using time-domain features and PSD-based features. For the space limit, we only exhibit the above 4 representative combinations. Asphalt and concrete are most likely to be misclassified to plastic and tile. Even in the best combination PSD-RF shown in Figure 7d, asphalt is misclassified to plastic for 23%, tile for 19% and concrete for

15% of the test cases, and concrete is misclassified to plastic for 46%, tile for 15% and asphalt for 18% of the test cases.



**Figure 6.** True positive rate for 8 terrain types with different classifiers using (a) Time-domain features, (b) FFT-based features, (c) PSD-based features.



**Figure 7.** Normalized confusion matrices of (a) T-NB, (b) T-RF, (c) PSD-NB, (d) PSD-RF.

### 5.2.2. Analysis of Performance of Feature-Learning Approaches

Now we are in the position to evaluate the performance of feature-learning approaches. The experiments are conducted on a system with a NVIDIA GeForce GTX 1080 Ti GPU and an Intel i7-7700K processor. We perform the experiments using different neural network model configurations to gain insight on the effect of learning spatiotemporal relationships. We consider the following five model configurations:

- **1D-CNN.** A one-dimensional convolutional neural network with alternating convolutional and pooling layers learns spatial features of a segment, and is followed by two fully connected layers and a SoftMax layer.
- **LSTM.** A single layer LSTM model directly handles 200 points length segments with 10 LSTM cells in series, which helps to learn temporal dynamics.
- **1DCL.** The proposed model is a spatiotemporal architecture.
- **1DCL-FC.** A variant of our 1DCL that we use a fully connected layer to replace the  $1 \times 1$  convolutional layer.
- **1DCL-3Conv.** Another variant of our 1DCL model that we consider increasing the number of convolutional layers to learn better spatial features. Here, we build two normal convolutional layers and a  $1 \times 1$  convolutional layer for a total of 3 convolutional layers.

Results from the comparison are shown in Table 2. The proposed 1DCL outperforms all the other models by achieving an accuracy of 80.18%, the best results of feature-engineering approaches by 10.30%, and the second method (LSTM) by 8.23%. The results demonstrate that the

spatiotemporal architecture we designed indeed learns essential characteristics and complex dynamics in proprioceptive signals. The CNN and LSTM achieve accuracies of 70.17% and 71.95%, which are both better than the best results of feature-engineering approaches. Hence, end-to-end learning methods can be employed to improve the performance and replace manually selecting features process in terrain classification field, which however, is rarely investigated currently.

**Table 2.** Performance of Feature-Learning Approaches.

Metrics	Networks				
	CNN	LSTM	1DCL	1DCL-FC	1DCL-3Conv
Accuracy	70.17%	71.95%	<b>80.18%</b>	67.94%	72.93%
F1-score	0.6893	0.7030	<b>0.7878</b>	0.6622	0.7159

The two variants of our 1DCL, 1DCL-FC, and 1DCL-3Conv, do not perform as well as the 1DCL seen from Table 2. Increasing the number of convolutional layers does learn better spatial features, yet increases the complexity of the model. Hence, the model is more likely to overfit. In contrast to  $1 \times 1$  convolution, fully connected layer has more parameters to train. Moreover,  $1 \times 1$  convolution keeps the structure of feature maps, in other words, each position of the output feature map of the  $1 \times 1$  convolutional layer corresponds to the same position of input feature maps. However, if replaced by a fully connected layer, the structure of feature maps is restructured, which may be the main reason for the relatively low accuracy of the 1DCL-FC.

### 5.2.3. Comparison of Classification Time

Since wheeled robot with a terrain-dependent control system works in outdoor environments, terrain classification must be implemented in real time to ensure the safety of the robot and make the robot adapt its driving style to the current terrain. As depicted in Figure 2, training process is done offline while prediction is done online, thus the time for classifying a test vector is more important. Training time is considered only in some situations when the trained model should be updated or retrained. Table 3 compares the computation times of different approaches. It is well known that the deep neural networks need a time-consuming training process. The training time of SVM is the highest in the feature-engineering approaches because the grid search to determine the optimal parameters is also very time-consuming. These computationally intensive training process can be done offline. As for online testing, using NB and DT, it takes only a few milliseconds. ELM, kNN, RF, and the three neural networks take no more than 40 milliseconds other than FFT-kNN. However, the test time of SVM and AdaBoost is significantly higher. On the other hand, in view of features, FFT and PSD take more time than time-domain features whatever training or testing.

**Table 3.** Running Time of Feature-Engineering and Feature-Learning Approaches.

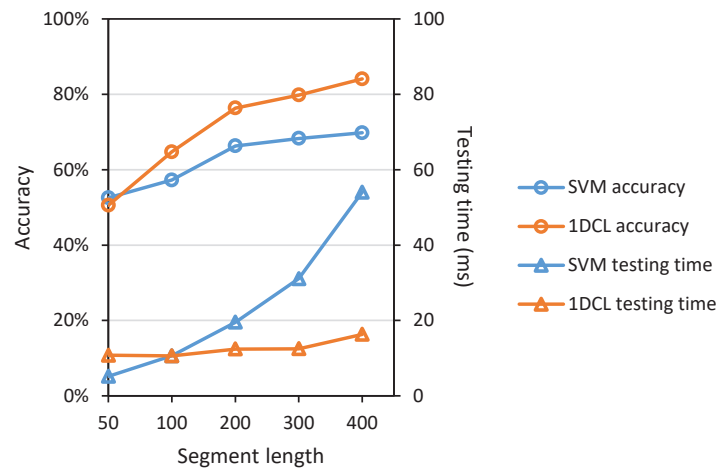
Classifiers	Training Time (s)			Testing Time (ms)		
	T	FFT	PSD	T	FFT	PSD
SVM	15.43	188.9	90.20	18.78	235.2	115.4
ELM	0.5709	1.401	0.5674	19.93	27.20	21.72
kNN	0.01219	0.01956	0.01423	8.298	69.40	36.63
NB	$8.120 \times 10^{-3}$	0.03499	0.01702	1.225	4.144	1.779
DT	0.01089	0.1596	0.08480	0.1409	0.4181	0.3472
RF	1.5371	22.80	11.81	16.35	10.71	9.431
Adaboost	6.911	173.1	82.26	81.54	96.39	82.07
CNN		251.8			7.110	
LSTM		746.5			13.18	
1DCL		1295			15.22	



#### 5.2.4. Comparison of Varying Segment Length

At the end of this section, we perform experiments using varying vibration signal segment length on two representative approaches: FFT-SVM and the proposed 1DCL. Please note that splitting raw vibration signal into segments with different lengths will result in different number of instances. Therefore, we randomly select the same number of instances for each experiment to ensure the comparability of the experiment results, which may correspondingly cause a slight drop for the accuracy due to the reduction in training samples.

Evaluating the segment sizes presents a resolution-efficiency trade-off. That is, a longer segment length contains more information of ground surface thus yielding increased accuracy requires more execution time, while a shorter segment length with faster real-time response leads to worse classification results. For applications of terrain classification, fast response and execution rates are essential for traversability evaluating. Figure 8 shows the classification accuracy at varying segment length and the corresponding testing time. It is observed that increasing the segment length does increase the accuracy. This growth slows down after the segment length exceeds 200. On the contrary, the testing time of SVM increases greatly with the segment length increasing. Interestingly, the testing time of the proposed 1DCL model does not seem to have a significant response to the varying segment length. Considering the segment length to classification time trade-off, we choose the segment length of 200 (i.e., 1 s) for the in-depth performance evaluation experiments as described above.



**Figure 8.** Accuracy and testing time of different approaches at varying vibration signal segment length. We take two representative approaches as examples: FFT-SVM and 1DCL.

## 6. Conclusions

In this paper, we conducted an extensive comparative study of VTC with shock absorbers. We presented a large body of experiments on feature-engineering approaches including 21 combinations of 3 kinds of features and 7 different classifiers, and feature-learning approaches including 3 neural networks. Referring to this study, one can find the most appropriate classification method according to their demands. If there are no limits on computational complexity, the proposed 1DCL could be the best solution. In future work, we will enhance the proposed 1DCL to implement real-time computation, which is indispensable for outdoor mobile robots, referring to literature [20]. Additionally, we will employ semi-supervised learning to solve the issues of the lack of labels in terrain classification.

**Author Contributions:** J.C. and W.L. proposed the 1DCL method; W.L. and Z.L. set up the experimental robot; X.L. collected and preprocessed the data; M.M., Y.L. and J.C. conceived, designed, and conducted the experiment; J.C., Y.L. and Z.L. analyzed the experiment results; M.M. and J.C. wrote the paper; X.L. polished the English writing.

**Funding:** This research was supported in part by the Educational and Scientific Research Projects for Young and Middle-Aged Teachers of Fujian Province (Grant No. JAT170934).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lozano-Perez, T. *Autonomous Robot Vehicles*; Springer Science & Business Media: Berlin, Germany, 2012.
2. García-Sánchez, J.; Tavera-Mosqueda, S.; Silva-Ortigoza, R.; Hernández-Guzmán, V.; Sandoval-Gutiérrez, J.; Marcelino-Aranda, M.; Taud, H.; Marciano-Melchor, M. Robust Switched Tracking Control for Wheeled Mobile Robots Considering the Actuators and Drivers. *Sensors* **2018**, *18*, 4316. [[CrossRef](#)] [[PubMed](#)]
3. Gonzalez, R.; Iagnemma, K. Slippage estimation and compensation for planetary exploration rovers. State of the art and future challenges. *J. Field Robot.* **2018**, *35*, 564–577. [[CrossRef](#)]
4. Helmick, D.; Angelova, A.; Matthies, L. Terrain adaptive navigation for planetary rovers. *J. Field Robot.* **2009**, *26*, 391–410. [[CrossRef](#)]
5. Iagnemma, K.; Dubowsky, S. Traction control of wheeled robotic vehicles in rough terrain with application to planetary rovers. *Int. J. Robot. Res.* **2004**, *23*, 1029–1040. [[CrossRef](#)]
6. Chen, Y.; Li, X.; Wiet, C.; Wang, J. Energy management and driving strategy for in-wheel motor electric ground vehicles with terrain profile preview. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1938–1947. [[CrossRef](#)]
7. Manjanna, S.; Dudek, G. Autonomous gait selection for energy efficient walking. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 5155–5162.
8. Zhao, X.; Dou, L.; Su, Z.; Liu, N. Study of the Navigation Method for a Snake Robot Based on the Kinematics Model with MEMS IMU. *Sensors* **2018**, *18*, 879. [[CrossRef](#)] [[PubMed](#)]
9. Zhu, Y.; Luo, K.; Ma, C.; Liu, Q.; Jin, B. Superpixel Segmentation Based Synthetic Classifications with Clear Boundary Information for a Legged Robot. *Sensors* **2018**, *18*, 2808. [[CrossRef](#)] [[PubMed](#)]
10. Khan, Y.N.; Komma, P.; Zell, A. High resolution visual terrain classification for outdoor robots. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 1014–1021.
11. Filitchkin, P.; Byl, K. Feature-based terrain classification for littledog. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012; pp. 1387–1392.
12. Anantrasirichai, N.; Burn, J.; Bull, D. Terrain classification from body-mounted cameras during human locomotion. *IEEE Trans. Cybern.* **2015**, *45*, 2249–2260. [[CrossRef](#)] [[PubMed](#)]
13. Ozkul, M.C.; Saranlı, A.; Yazicioglu, Y. Acoustic surface perception from naturally occurring step sounds of a dexterous hexapod robot. *Mech. Syst. Signal Process.* **2013**, *40*, 178–193. [[CrossRef](#)]
14. Christie, J.; Kottege, N. Acoustics based terrain classification for legged robots. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 3596–3603.
15. Valada, A.; Burgard, W. Deep spatiotemporal models for robust proprioceptive terrain classification. *Int. J. Robot. Res.* **2017**, *36*, 1521–1539. [[CrossRef](#)]
16. Wu, X.A.; Huh, T.M.; Mukherjee, R.; Cutkosky, M. Integrated Ground Reaction Force Sensing and Terrain Classification for Small Legged Robots. *IEEE Robot. Autom. Lett.* **2016**, *1*, 1125–1132. [[CrossRef](#)]
17. Walas, K. Terrain classification and negotiation with a walking robot. *J. Intell. Robot. Syst.* **2015**, *78*, 401–423. [[CrossRef](#)]
18. Hoffmann, M.; Štěpánová, K.; Reinstein, M. The effect of motor action and different sensory modalities on terrain classification in a quadruped robot running with multiple gaits. *Robot. Auton. Syst.* **2014**, *62*, 1790–1798. [[CrossRef](#)]
19. Weiss, C.; Tamimi, H.; Zell, A. A combination of vision- and vibration-based terrain classification. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 2204–2209.
20. Otte, S.; Weiss, C.; Scherer, T.; Zell, A. Recurrent Neural Networks for fast and robust vibration-based ground classification on mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 6–20 May 2016; pp. 5603–5608.

21. Bermudez, F.L.G.; Julian, R.C.; Haldane, D.W.; Abbeel, P.; Fearing, R.S. Performance analysis and terrain classification for a legged robot over rough terrain. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012; pp. 513–519.
22. Libby, J.; Stentz, A.J. Using sound to classify vehicle-terrain interactions in outdoor environments. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 3559–3566.
23. Hoepflinger, M.A.; Remy, C.D.; Hutter, M.; Spinello, L.; Siegwart, R. Haptic terrain classification for legged robots. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, Alaska, 3–8 May 2010; pp. 2828–2833.
24. Kurban, T.; Beşdok, E. A comparison of RBF neural network training algorithms for inertial sensor based terrain classification. *Sensors* **2009**, *9*, 6312–6329. [[CrossRef](#)] [[PubMed](#)]
25. Khan, Y.N.; Masselli, A.; Zell, A. Visual terrain classification by flying robots. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 498–503.
26. Yin, J.; Yang, J.; Zhang, Q. Assessment of GF-3 polarimetric sar data for physical scattering mechanism analysis and terrain classification. *Sensors* **2017**, *17*, 2785. [[CrossRef](#)] [[PubMed](#)]
27. Yan, Y.; Rangarajan, A.; Ranka, S. An Efficient Deep Representation Based Framework for Large-Scale Terrain Classification. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 940–945.
28. Gonzalez, R.; Iagnemma, K. DeepTerramechanics: Terrain Classification and Slip Estimation for Ground Robots via Deep Learning. *arXiv* **2018**, arXiv:1806.07379.
29. Lu, L.; Ordóñez, C.; Collins, E.G.; DuPont, E.M. Terrain surface classification for autonomous ground vehicles using a 2D laser stripe-based structured light sensor. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 11–15 October 2009; pp. 2174–2181.
30. Rankin, A.; Huertas, A.; Matthies, L.; Bajracharya, M.; Assad, C.; Brennan, S.; Bellutta, P.; Sherwin, G.W. Unmanned ground vehicle perception using thermal infrared cameras. *Unmanned Systems Technology XIII. Int. Soc. Opt. Photonics* **2011**, *8045*, 804503.
31. Zhou, S.; Xi, J.; McDaniel, M.W.; Nishihata, T.; Saleses, P.; Iagnemma, K. Self-supervised learning to visually detect terrain surfaces for autonomous robots operating in forested terrain. *J. Field Robot.* **2012**, *29*, 277–297. [[CrossRef](#)]
32. McDaniel, M.W.; Nishihata, T.; Brooks, C.A.; Saleses, P.; Iagnemma, K. Terrain classification and identification of tree stems using ground-based LiDAR. *J. Field Robot.* **2012**, *29*, 891–910. [[CrossRef](#)]
33. Brooks, C.A.; Iagnemma, K. Vibration-based terrain classification for planetary exploration rovers. *IEEE Trans. Robot.* **2005**, *21*, 1185–1191. [[CrossRef](#)]
34. Weiss, C.; Frohlich, H.; Zell, A. Vibration-based terrain classification using support vector machines. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 4429–4434.
35. Chen, Z.; Li, C.; Sanchez, R.V. Gearbox fault identification and classification with convolutional neural networks. *Shock Vib.* **2015**, *2015*. [[CrossRef](#)]
36. Ince, T.; Kiranyaz, S.; Eren, L.; Askar, M.; Gabbouj, M. Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks. *IEEE Trans. Ind. Electron.* **2016**, *63*, 7067–7075. [[CrossRef](#)]
37. Abdeljaber, O.; Avci, O.; Kiranyaz, S.; Gabbouj, M.; Inman, D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* **2017**, *388*, 154–170. [[CrossRef](#)]
38. Wang, S.; Kodagoda, S.; Shi, L.; Dai, X. Two-Stage Road Terrain Identification Approach for Land Vehicles Using Feature-Based and Markov Random Field Algorithm. *IEEE Intell. Syst.* **2018**, *33*, 29–39. [[CrossRef](#)]
39. Otsu, K.; Ono, M.; Fuchs, T.J.; Baldwin, I.; Kubota, T. Autonomous terrain classification with co-and self-training approach. *IEEE Robot. Autom. Lett.* **2016**, *1*, 814–819. [[CrossRef](#)]
40. Dutta, A.; Dasgupta, P. Ensemble learning with weak classifiers for fast and reliable unknown terrain classification using mobile robots. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 2933–2944. [[CrossRef](#)]
41. Zhao, K.; Dong, M.; Gu, L. A New Terrain Classification Framework Using Proprioceptive Sensors for Mobile Robots. *Math. Probl. Eng.* **2017**, *2017*, 3938502. [[CrossRef](#)]

42. Vicente, A.; Liu, J.; Yang, G.Z. Surface classification based on vibration on omni-wheel mobile base. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 916–921.
43. Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufier, M.; Verstockt, S.; Van de Walle, R.; Van Hoecke, S. Convolutional neural network based fault detection for rotating machinery. *J. Sound Vib.* **2016**, *377*, 331–345. [[CrossRef](#)]
44. Weiss, C.; Fechner, N.; Stark, M.; Zell, A. *Comparison of Different Approaches to Vibration-Based Terrain Classification*; EMCR: Freiburg, Germany, 19–21 September 2007.
45. Komma, P.; Weiss, C.; Zell, A. Adaptive bayesian filtering for vibration-based terrain classification. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'09, Kobe, Japan, 12–17 May 2009; pp. 3307–3313.
46. Zou, Y.; Chen, W.; Xie, L.; Wu, X. Comparison of different approaches to visual terrain classification for outdoor mobile robots. *Pattern Recognit. Lett.* **2014**, *38*, 54–62. [[CrossRef](#)]
47. Bouguelia, M.R.; Gonzalez, R.; Iagnemma, K.; Bytner, S. Unsupervised classification of slip events for planetary exploration rovers. *J. Terramech.* **2017**, *73*, 95–106. [[CrossRef](#)]
48. Chang, C.C.; Lin, C.J. LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol. (TIST)* **2011**, *2*, 27. [[CrossRef](#)]
49. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [[CrossRef](#)]
50. Nguyen, B.P.; Tay, W.L.; Chui, C.K. Robust biometric recognition from palm depth images for gloved hands. *IEEE Trans. Hum.-Mach. Syst.* **2015**, *45*, 799–804. [[CrossRef](#)]
51. Domingos, P.; Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Mach. Learn.* **1997**, *29*, 103–130. [[CrossRef](#)]
52. Breiman, L. *Classification and Regression Trees*; Routledge: London, UK, 2017.
53. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
54. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [[CrossRef](#)]
55. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)] [[PubMed](#)]
56. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
57. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
58. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
59. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 2951–2959.
60. Yang, Y.; Liu, X. A re-examination of text categorization methods. In Proceedings of the 22nd International Conference on Research and Development in Information Retrieval, Berkeley, CA, USA, 15–19 August 1999.